

SW Functional Description

Chai, Chung Shien

Last updated: 9 March 2019

First created: February 2013

Intel Corporation Confidential

Table of Contents

[1 Introduction 3](#_Toc2901099)

[1.1 Overview 3](#_Toc2901100)

[1.2 User flow 5](#_Toc2901101)

List of Figures

List of Tables

# Introduction

The tool is requested to sign PAC data. Refer <doc link> for the detail of PAC Authentication format.

## Overview

The signing tool currently only support signing key from openssl, which is in “.pem” extension (Privacy Enhanced Mail Certificate).

Signing keychain is in “.qky” extension (Quartus Key) *[This is same as Stratix 10 keychain file, however the format of the keychain of Stratix10 and PAC are different, should we think a new file type name?!?!?]*

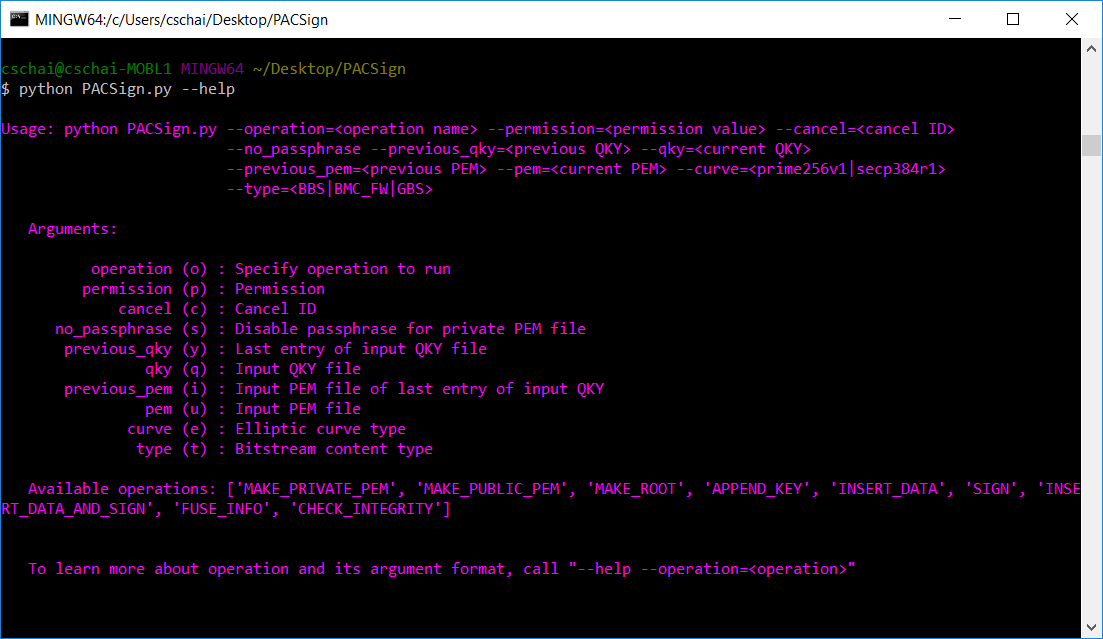
You must use Python 3 to execute the script

In general, the format to call signing tool is

python PACSign.py --operation=<operation to peform> [other argument(s)]

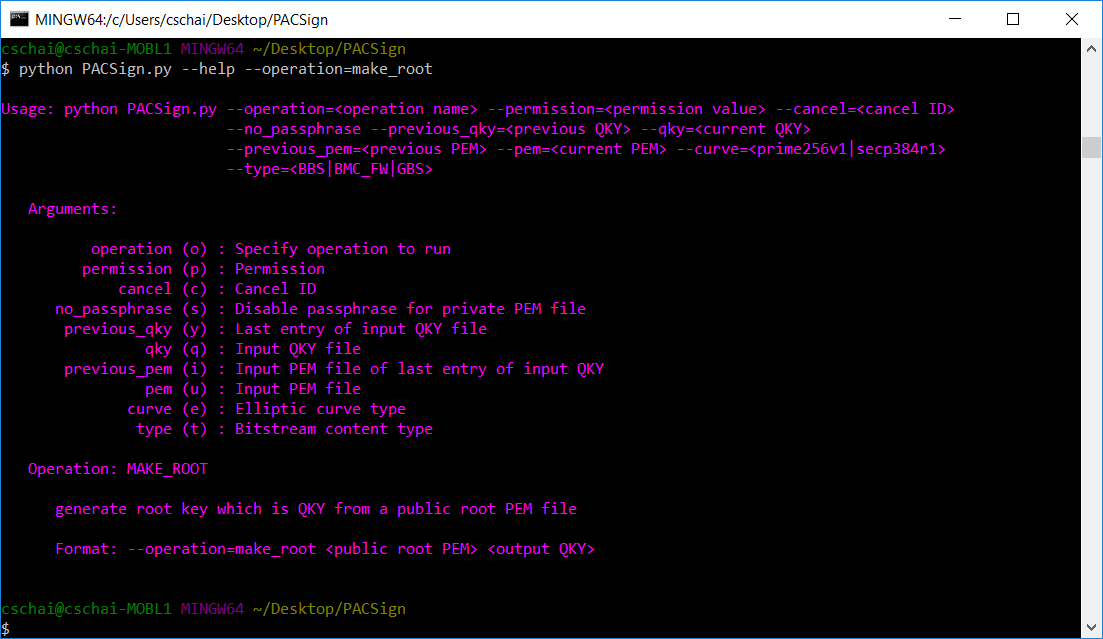
Use help menu to learn available operations.

python PACSign.py --help



Use help menu to further learn each of the available operations.

python PACSign.py --help --operation=<operation name>



## User flow

Here are the steps to sign PAC data.

1. Generate a pair of private/public key to create the Root keychain
2. Operation “make\_private\_key” is used to generate private key
3. Operation “make\_public\_key” is used to generate public key from private key
4. Operation “make\_root” is used to generate Root keychain from public key
5. Append new public key into the root keychain to create the final keychain
6. Generate new pair of private/public key
7. Operation “append\_key” is used to append the new public key into Root keychain and generate final keychain
8. Sign PAC data. There are two ways to sign the raw PAC data
   * Sign the raw data in one step.
     1. Operation “insert\_data\_and\_sign” is used to auto-prepend Block0 and Signature Block (or Block1) into raw data and sign it with the keychain.
   * Sign the raw data in two separate steps.
     1. Operation “insert\_data” is used prepend Block0 and Signature Block (or Block1) into raw data. The output is an unsigned data.
     2. Operation “sign” is used to sign the unsigned data
9. Check fuse information
10. Operation “fuse\_info” is used to print the fuse information into text
11. Check data integrity
12. Operation “check\_integrity” is used to check whether data is signed and signature chain integrity. This operation is not used to check the raw data.